Vision 2000: Radical Reengineering of the Hubble Space Telescope Control Center System

Douglas R. Spiegel, NASA/Goddard Space Flight Center, doug.spiegel@gsfc.nasa.gov
Larry Barrett, Orbital Sciences Corporation, larry.barrett@gsfc.nasa.gov

Vision 2000 Collocation Facility
9701 Philadelphia Way
Lanham, MD 20706
fax: (301) 918-7474

ABSTRACT

The Hubble Space Telescope operations control center ground systems are undergoing a radical reengineering effort to improve efficiencies, reduce costs, and provide unprecedented engineering support to both scientists and engineers as the Hubble mission continues well into the next century. This paper discusses the new architecture that integrates the processes of real-time monitoring, engineering data archiving, data analysis and trending, and spacecraft commanding. It discusses how the implementation of Java and state-of-the-art security tools means that Hubble engineering data can be accessed easily from anywhere in the world while maintaining the integrity of the command and control systems. This architecture is uniquely scaleable: from a single processor supporting a few users to the operational system capable of supporting hundreds of geographically dispersed users during the Hubble's planned servicing missions. Automatic failover of all critical hardware and software processes will eliminate the need for constant manual monitoring of the ground system hardware and networks.

The new system includes several key breakthroughs in flexibility and end-user customization which increase the productivity of the flight operations team. Using Java applets, users can create customized telemetry display pages instantly and easily. A new data archive interface allows users to request any data from any time period in the Hubble's history and receive it in minutes, without having to know how the data is stored. Automated real-time fault detection will permit the flight operations team to spend time in flight system analysis and anomaly resolution, rather than tedious data monitoring. Scientists can access engineering data directly to gain critical insight into the conditions of the Telescope while it was making a science observation.

INTRODUCTION

NASA's Hubble Space Telescope (HST) has been producing extraordinary scientific results since its launch in April 1990, enhanced with improved Science Instruments and replacement hardware from two successful on-orbit Servicing Missions. In order to maximize the science return and ensure the health and safety of the vehicle for an extended mission lifetime, the HST Project recognized that significant change was needed to improve the spacecraft operations and maintenance of the ground system. In 1995, the HST Project initiated the Vision 2000 effort to modernize and streamline the ground systems. Vision 2000 represented a challenge to reduce mission costs while vastly improving efficiencies and overall engineering support, without introducing increased risk to the spacecraft or the science program. The largest component of the ground system, the Control Center System (CCS), underwent a radical reengineering effort to replace the set of disparate systems that had been designed and developed in the late 70s/early 80s. In short, a world-class ground system for a world-class science observatory. This paper describes the approach, key technologies and architecture components selected to revolutionize HST flight and servicing operations, along with spacecraft hardware/flight software integration and test (I&T).

APPROACH

In undertaking this endeavor, which had to be accomplished in the midst of ongoing operational support as well as preparation and execution of the 2nd Servicing Mission (SM), the project had to challenge the existing business processes to optimize spacecraft operations. The HST project chartered an integrated Product Development Team (PDT) to reengineer the processes and systems that comprise the CCS. The CCS PDT was established as a “badgeless” team comprised of civil service and multi-contractor staff, collocated in a single facility in Lanham, Maryland. A key component of the PDT approach was to integrate domain and technology experts, including the users, developers, testers, network and security engineers and system integrators, into a cohesive team. A major challenge facing the PDT was to incorporate new technology, integrate commercial-off-the-shelf (COTS) products with legacy software, and provide access to the system from anywhere in the world. In addition, the new ground system had to be very scaleable, permitting its deployment at a number of disparate test and integration sites as well as the operational facilities (over 20 instances).

The CCS PDT began by evaluating new technologies and COTS products, and prototyping those that showed promise. Concurrently, key spacecraft engineers, flight operations personnel, and system designers undertook a business process reengineering (BPR) activity to first characterize the current processes and then to devise the “new way of doing business.” The resulting future state models guided the development of the CCS architecture. As the transition from BPR to system design engaged, the prototyping activities culminated in the demonstration of an integrated proof-of-concept system. This served as an excellent vehicle to communicate the concepts and architectural elements being explored, and allowed the users to provide meaningful feedback early in the process.

Another challenge facing the PDT was to develop the system within an aggressive schedule constrained to support preparation for the third SM. This, coupled with the major goal of reducing maintenance costs, led to the need for a streamlined and efficient yet comprehensive development approach. A hybrid methodology based on Object-Oriented Analysis and Design (OMT), Structured Analysis and Design (Gane & Sarsen), Information Engineering (J. Martin), and rapid prototyping, was employed to meet the differing needs of various components of the system. An important point here was not to constrain development with a mandated “one size fits all” approach. The system specification, design data, and test information were captured in a CASE tool environment to support development and future maintenance. The new software was developed in Java, C++, Tcl, SQL, C, and scripting languages (e.g., csh, perl). In addition, portions of the legacy Fortran code had been ported (e.g., command bit construction) where it was determined that a COTS solution was not feasible and that rewriting the software was not cost-effective. The system was developed in incremental releases, with each release being deployed into “shadow-mode” (or semi-) operational use for evaluation and testing by the users. The PDT has delivered six major releases of CCS in just 18 months, demonstrating the value of collocation and the badgeless team.

SYSTEM ARCHITECTURE

The CCS is a major component of the overall HST ground system, which operates in conjunction with the Planning & Scheduling System (P&S), the Science Data Processing System (SDP), and Flight Software (FSW) (the other PDTs that comprise Vision 2000 project). The CCS, operated at the Goddard Space Flight Center in Greenbelt, Maryland, performs spacecraft commanding, communications management, and engineering data analysis for health and safety management.

The design and implementation of the CCS architecture has presented several unique engineering challenges. The most ambitious of these was the specification of a common architecture that was scaleable in its use of processing resources, its inherent functionality, and the size of the user community it could support. For example, the smallest system configuration must execute on a single, stand-alone computer and provide only essential spacecraft commanding, telemetry ingest, and basic analysis functions. The operational configuration, which is used to control the spacecraft, is hosted on several multi-processor computers with full-scale functionality including: coordination of spacecraft communications, near-real-time monitoring of spacecraft health and safety, management of a multi-terabyte engineering data archive, and extensive engineering data analysis functions.

A secondary challenge was to support security, network, and system management functions across these scaled systems while providing simplified user access methods through the use of intranet (i.e., web-based) technologies. These non-cooperative technologies had to be balanced to provide adequate system security mechanisms and to protect the vehicle command and control functions at the core of the system. This had to be accomplished without impeding other system infrastructure elements, and without constraining the engineers and scientists who need to access vehicle parameter data and analysis products from off-site workstations.

An overview of the system context is depicted in Figure 1, which illustrates the connectivity to the operational flight and ground systems as well as the integration and test environments. CCS receives planned spacecraft events and command loads from the P&S at the Space Telescope Science Institute (STScI) in Baltimore, Maryland, based on the upcoming science observations, which are uplinked to the spacecraft. Engineering telemetry data is received and processed by CCS, which then provides the data in real-time to client workstations and also stores the data in a life-of-mission archive. This data is also provided to the SDP (at STScI) for use in calibration and analysis of science observations. CCS interacts with flight software facilities to uplink and validate the contents of the spacecraft's onboard computer programs.

The CCS is a data-driven, multi-server architecture that is segmented into security levels to accommodate remote engineering data access while protecting the command and control systems (see Figure 2). Firewalls, network routers and information security functions have been integrated to provide the necessary filtering, authentication, access controls and auditing. An early decision to design and build-in the security mechanisms from the start of the project has been very beneficial, having actually facilitated development of a highly modular system.

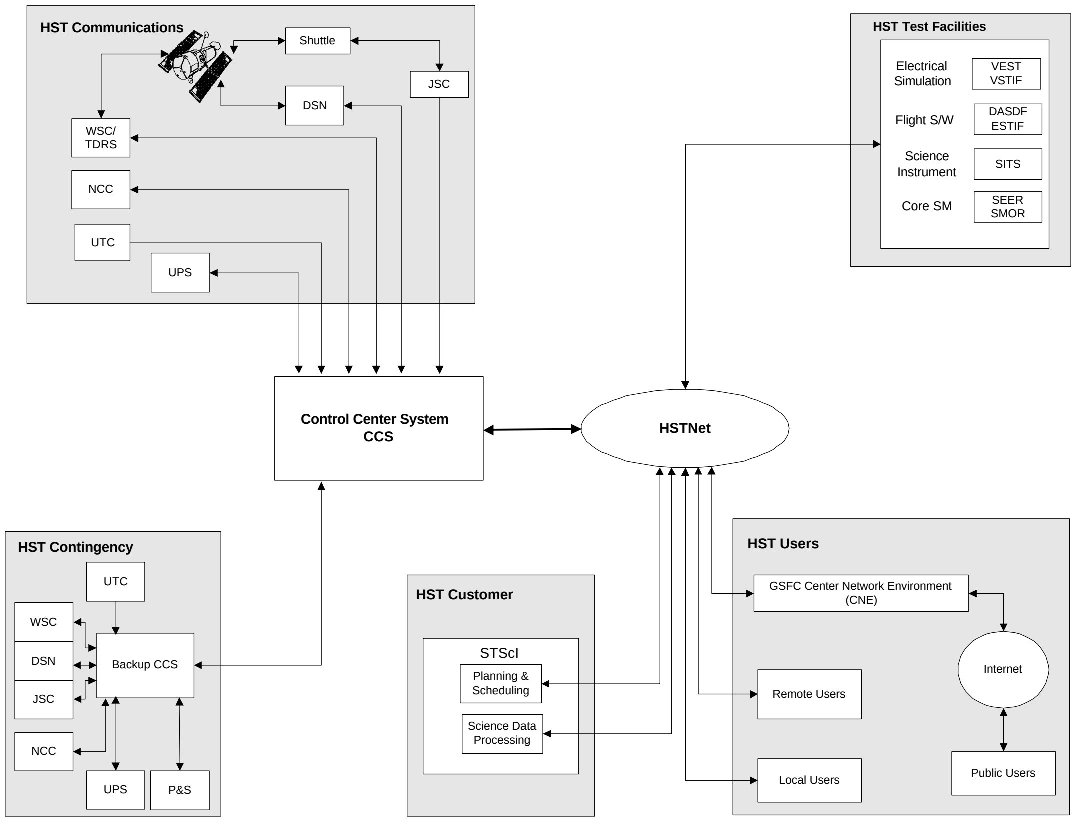

Figure 1. CCS Context Diagram

The CCS is partitioned into six major subsystems: Graphical User Interface (GUI), Command Processing, Front-end Processing (FEP), Systems Monitoring, Data Management, and CCS Management. An architectural cornerstone for these subsystems was the decision to employ a middleware layer for inter-process communication. As the foundation for the CCS software, middleware provides a suite of services for message and data transport between disparate applications and COTS products, executing on heterogeneous hardware platforms.

The PDT has integrated over 20 COTS (and Government OTS, *GOTS*) products and developed over 500,000 source lines of code (LOC), which has been configured to run in both the full operational multi-server and the scaled-down single-server configurations. A discussion of the key technologies, COTS products, and functionality for each subsystem is provided in the following paragraphs.

Graphical User Interface - One of the major results from the early prototyping activities was the discovery that *Java* technology could be used in a mission-critical environment and yet provide the needed flexibility and power to support remote access analysis tools. The entire CCS GUI is implemented in the Java programming language, which has proven to be a viable, highly productive, platform-independent software technology. Since the Java code executes within the Java Virtual Machine (VM) resident on the client workstation, it can be configured as either “applets” or standalone applications. The GUI can run on a variety of Unix workstations and

Windows (NT, 95) PCs either over the internet (browser interface) or intranet (network interface). CCS has found that standard configuration PCs/NT serve well for reliable, fast, low-cost user workstations. Through the user's desktop PC in their office, or other remote location, all CCS tools and functions are available (except commanding which is protected). For the user, CCS becomes another tool in the desktop tool suite and coexists with email, web-browsing, office applications, etc.

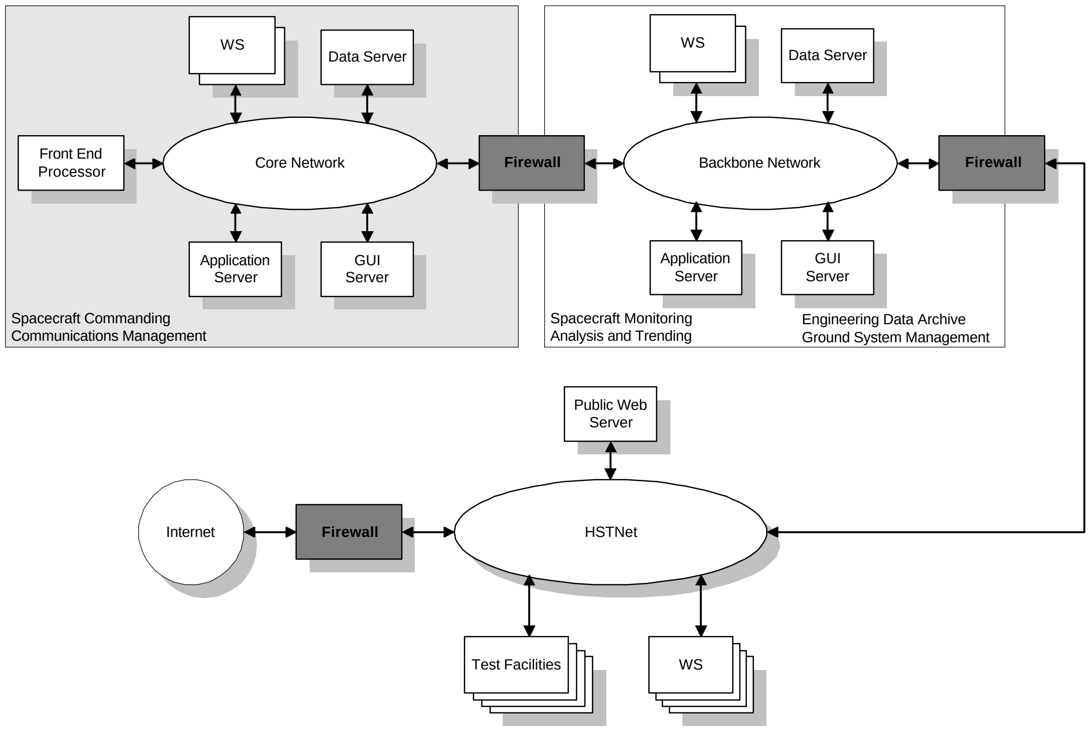

Figure 2. CCS System Architecture Overview

The CCS client-server architecture enables the multi-processor servers to handle the resource-intensive tasks and handling of large data volumes, while the clients perform graphics rendering and tasks such as local printing. By implementing the GUI in Java, many tasks can be performed locally without significant loading on the network, while still providing a thin client (Java applets download on a demand basis). Also, by performing the data processing and computations on the server side, and distributing changes-only data on a demand-only basis, the amount of data flowing to the client workstation is relatively small, minimizing network traffic.

The GUI provides an integrated suite of tools and displays. The Mission Engineer (ME) toolbox provides access to the commanding functions, mission timeline scheduling/control, real-time data display page builder/viewer, historical engineering data analysis, spacecraft monitoring and calibration functions, viewing/filtering CCS system events, and control and monitoring of the CCS ground system. In addition, a toolset is provided that supports user collaboration, allowing ME's to view and annotate system products simultaneously and communicate via whiteboarding, chat, audio, and video-conferencing, all with their desktop PCs.

A powerful tool used extensively by the ME's allows them to build and view customized real-time display pages instantly and easily. These display pages are used to monitor the status of all onboard components, communications links, and ground system status (over 6000 flight parameters and several thousand ground system parameters). The display page is treated as an blank palette onto which the user can add components such as stripcharts, telemetry mnemonic values, limit monitors, etc., and organize them any way they wish. The user can "stack" sets of components on the page via tab groups, which allows the user to cycle through the sets of mnemonics without having to call up separate pages (some of the operations pages contain over 700-800 mnemonics each). The displays automatically establish a connection to the real-time data stream, even when building a new custom page. With the Java/Internet technology, users can view these pages from any location (office, home, on the road), and have instant access to the real-time engineering data.

Commanding - Spacecraft commanding is the most critical function of the ground system since, without it, HST could not accomplish its mission. Hence, the security requirements are strict, and the software is very specific to the spacecraft (no COTS products could be obtained to provide the core commanding functions). However, the event scheduling and execution functions needed to automate commanding are more generic. The PDT was able to identify and integrate a database-oriented GOTS product to manage the spacecraft event data and the mission timeline. The Mission Operations Planning and Scheduling System (MOPSS), developed by and supporting several missions at GSFC, provides a powerful graphical user interface that allows users to select views of the spacecraft timeline (HST has several dozen event types), and then manipulate and possibly reschedule those events (e.g., communications satellite service window changes just before planned contact). Next, an execution engine, which is fed the approved operational timeline of events, processes each event (e.g., command load uplink). A knowledge base specifies rules to direct the execution engine in the handling of anomalous events (e.g., lost contact before completion of uplink). The execution engine will initially operate in an "advisory" mode, to allow testing of the knowledge base and building confidence in the system's accuracy.

Front-end Processing - The FEP is the communication interface for telemetry, command, and external communications management. A COTS hardware/software product, VEDA Omega series, processes the incoming telemetry data including removing communications protocol artifacts, time-tagging, de-commutation and conversion of the raw data to engineering units. The FEP processes command data by wrapping it in the appropriate communications protocol and routing to the target destination (HST operates in several modes for nominal operations, servicing missions, contingency, and testing purposes). The FEP is a data-driven system which automatically detects the presence and format of the telemetry data and eliminates the need for manually loading catalogs, a necessity in the current system. In addition to routing the processed telemetry data to the archive, the FEP distributes this data in real-time for display and monitoring purposes. A GOTS product, the Information Sharing Protocol (ISP), provided by Johnson Space Center, provides a real-time data distribution infrastructure, which allows CCS applications to publish and subscribe to data of interest. The data is distributed in a changes-only format, which minimizes the network traffic and volume of data coming into the user's workstation. The FEP's interfaces to the other CCS components are isolated via the middleware.

System Monitoring - One of the primary Vision 2000 goals was to provide unprecedented engineering support to both scientists and engineers. The System Monitoring subsystem provides an engineering data analysis toolset for general plotting, trending, reporting, and analysis. The automated real-time monitoring component performs fault detection and isolation, working in conjunction with the command and telemetry components to predict spacecraft state, compare to actual state, and analyze miscompares to determine if an anomaly has occurred or if performance is degrading. A knowledge base contains state-based rules for isolating a condition given an appropriate combination of parameters. The results of the analysis (for recovery of a potential fault) are currently handled by personnel notification (pager, email). The CCS PDT is assessing the value of automating the fault recovery process, which would involve closed-loop commanding (approaching *"lights-out"* operations). COTS products have been integrated for the knowledge base and inference engine (RTworks), and data plotting and analysis (PV-Wave).

System Monitoring provides a general service for the generation, collection and distribution of CCS system events. These events are notices of processing activities, error conditions, alerts, etc. that are displayed in real-time, and logged for historical analysis. This data is useful in analyzing spacecraft anomalies and troubleshooting ground system problems.

System Monitoring also provides spacecraft sensor analysis tools, which perform attitude determination, sensor bias and calibration computations, and sensor data analysis algorithms (e.g., FFT). These functions are provided by ported legacy Fortran software that has been integrated into the CCS architecture. The decision to port this software was driven by the complexity of the algorithms, the stability of the software, and the projected cost of rewriting these functions.

Data Management - The Data Management subsystem provides data storage, cataloging, retrieval, and database services for the entire system. The engineering data archive, due to the vast volume of data (~ 4GB/day), proved to be a significant design challenge. A balance had to be reached between accurately archiving all of HST's telemetry data (over 6000 parameters in a 32 kilobits per second stream) and providing engineers the ability to quickly and easily access the data going all the way back to launch (15 years, ~15 TB). To meet this demand, the engineering data archive was designed with three major components: an all-points, flat-file archive; an online *data warehouse* (a new technology in data storage/retrieval for rapid complex query support); and a data server that provides access to all of the data in a seamless manner.

The captured engineering data, both real-time and onboard recorder playback, is processed into a seamless stream containing all points of all downlinked parameters. This "merged" data, or all-points, is kept online (RAID) for 30-days and is archived for the life-of-mission using an DLT tape archive management system (AMASS). The data warehouse, the Redbrick COTS product, is loaded with a filtered set of the all-points data (changes-only and averages) for the life of the mission and is kept entirely online for rapid retrieval (~ 1.5TB). The data warehouse product augments the archive by providing both rapid access and complex query support, which cannot be reasonably achieved with flat files. Designed for *load-once, retrieve many* operation, the data warehouse stores the data using a bit-mapped indexing scheme that provides powerful query capability and excellent performance, even for retrievals covering large time spans (ME's trend data for the life of the mission to analyze anomalies and component performance). The Data Server (custom software) insulates the user and other CCS applications from the details of data storage, providing the flexibility to transparently tune the file structures and other physical properties.

The Data Management subsystem works in conjunction with the GUI and System Monitoring to support a comprehensive, integrated engineering analysis toolset by providing services to perform queries, data extraction, data storage, reference data retrieval, and event logging. The system reference data is stored in an Oracle RDBMS (relational database management system), with standard APIs (application program interfaces) to allow applications access to the data.

CCS Ground System Management - In managing the overall ground system to provide system configuration, startup/shutdown control, monitoring and failover, several COTS products were integrated with a custom server. CCS monitors itself at the network, hardware, and software application levels to determine its state, and to aid in the detection and resolution of problems. The network elements are monitored with HP Openview. The system hardware (cpu, memory, disk) and applications are monitored by a combination of COTS products (Patrol) and custom software. Information generated from these sources is collected by a knowledge-based product (Tivoli T/EC) that has been programmed to interpret the events and determine appropriate action (e.g., restart a process, failover to another node, notify a system administrator). These ground system management functions have been designed to be scaleable across the span of CCS configurations without requiring software modification.

CONCLUSION

The HST Project has been successful at challenging the past methods of spacecraft operations and developing a state-of-the-art ground system. Using an integrated PDT and hybrid development methodology allowed the team to deliver a high-quality, secure, modular architecture that provides the desired level of remote access for engineering data analysis.

One of the key achievements has been the delivery of a system that is scaleable, without software modification, from a single-server configuration capable of supporting a few users to the operational system capable of supporting hundreds of users during the Hubble's planned servicing missions. The architecture allows access from anywhere in the world while protecting the command and control functions of the world's leading scientific on-orbit observatory.